Technical manual

Application Outcomes

1. Build a base class for Differentiated Services in ns-3.
2. Implement Strict Priority Queueing (SPQ) and Deficit Round Robin (DRR), using the base class.
3. Validate and verify your SPQ and DRR implementations.

Components

DiffServ

A Base Class to Simulate Differentiated Services

Functionality: classification - The classify function utilizes filter aspect to sort the traffic packets into appropriate traffic queues; scheduling - The schedule function carries out designed Quality-of-Service algorithm to schedule which traffic queue to be served at the time.

Strict Priority Queueing

Classifies network packets as either priority or regular traffic and ensures that priority traffic will always be served before low priority. Priority packets and regular packets are filtered into separate FIFO queues, the priority queue must be completely empty before the regular queue will be served. The advantage of this method is that high priority packets are guaranteed delivery so long as their inflow does not exceed the transmission rate on the network. The potential disadvantage is a high proportion of priority traffic will cause regular traffic to suffer extreme performance degradation

Deficit Round Robin

Classifies packets into different queues, associates a fixed quantum to each queue. A deficit counter is used to keep track of the credit available to each queue. The variable Quantum is the number of bytes that each queue can transmit in each turn. The Deficit Counter is used to keep track of the credit (deficit) available to each queue. Each queue is allowed to send a given amount of bytes (Quantum) in each round of the robin. If the packet size at the front of the queue is larger than the amount of the Quantum, then the queue will not be serviced.

Validation

SPQ: two client server applications A and B. Router on the path is configured as A traffic is prioritized over B. Application A begins sometime after application B starts.

DRR: three queues with quanta ratio 3:2:1. Three client-server bulk data transfer applications will start at the same time.

Installation

Prerequisites

Installation is divided into parts

- Part 1: Install

Virtual Box (Oracle), version 6.0.
Install Ubuntu, version 18.04.01.

- Part 2: Prerequisites of ns3 installation

C++ compiler, Python, Git, tar, bunzip2

- Part 3: Installtest NS-3 using the link:

https://www.nsnam.org/docs/tutorial/html/getting-started.html

Program Installation

It is important to clone the repository from GitHub:
git clone https://github.com/BettyRain/cs621.git
cd cs621
./waf configure –build-profile=debug –enable-examples –enable-tests
./waf build

- To run DRR simulation:

./waf –run drr-runner

- To run SPQ simulation:

./waf –run spq-runner

Changed/Created	Components	Usage
ns3-dev/src/traffic-control/model	Base class and queues	
diff-serv.[cc/h]	queue base class	serves as the base class for the custom queues being implemented. Offers basic functionality that is the same across all queues and allows custom queues to be used with p2p-net-device through inheriting queue<item>
traffic-class.[cc/h]	queue component	the queue of packets for the queue classes to use. each TC represents one priority level queue and keeps track of the number of bytes and packets that it has. Also contains a filter to make it easy for the queue class to identify which queue packets belong in
filter.[cc/h]	queue component	this is a way to contain filter elements in an easy to manage
filter-element.[cc/h]	queue component	
drrqueue.[cc/h]	queue	deficit round robin queue implementation
strict-priority-queue.[cc/h]	queue	string priority queue implementation
ns3-dev/scratch/	Simulation and verification	
spq-runner.cc	SPQ validation scenario	A traffic is prioritized over B
drr-runner.cc	DRR validation scenario	three queues with quanta ratio 3:2:1
cs621/ns-3-dev/	Config files and Pcaps	
spq-config.txt	SPQ config	In that file you should specify the number of queues and a priority level for each queue
drr-config.txt	DRR config	It specifies the number of queues and the corresponding quantum value for each queue
pre-SPQ-0-0.pcap	SPQ pre pcap	It shows a number of only low priority packets first, followed by interleaving high and low priority packets).
post-SPQ-1-1.pcap	SPQ post pcap	It shows a number of only low priority packets first, followed by only high priority packets, followed by only low priority packets.
pre-DRR-0-0.pcap	DRR pre pcap	It shows the interleaving three traffic streams close to a 123123123 pattern, if 1, 2, 3 represent the packets in each traffic stream.
post-DRR-1-1.pcap	DRR post pcap	It shows a pattern of packets closely aligned with the respective quanta (weights).

Table 1: Files used in program, relation to components and usage description